Lab2 Review

NTHU Logic Design Laboratory

2019/10/15

By Prof. Chun-Yi Lee

Outline

1. Basic - NOR gates only
2. Advanced - Carry-Lookahead Adder
3. FPGA - 4-bit CLA
4. Lab Rule

Outline

1. Basic - NOR gates only
2. Advanced - Carry-Lookahead Adder
3. FPGA - 4-bit CLA
4. Lab Rule

Basic - NOR gates only (1/2)

Verilog Question 1

- (Gate Level) NOR gates only
  - Use **NOR gates only** to realize the following functions
  - **NOT, NOR, AND, OR, XOR, XNOR, NAND**
  - Input/Output: A (1bit), B (1bit), Sel (3 bits), Out (1 bit)
  - **Please draw your circuits in your report**

Sel [2:0]	Out
000	Out = !A
001	Out = A nor B
010	Out = A and B
011	Out = A or B
100	Out = A xor B
101	Out = A xnor B
110 & 111	Out = A nand B

8

Basic - NOR gates only (2/2)

- 大部分的人沒有提到如何使用 nor gate 實作 MUX
- 很多人沒有畫出完整的的電路圖

Outline

1 Basic - NOR gates only

2 Advanced - Carry-Lookahead Adder

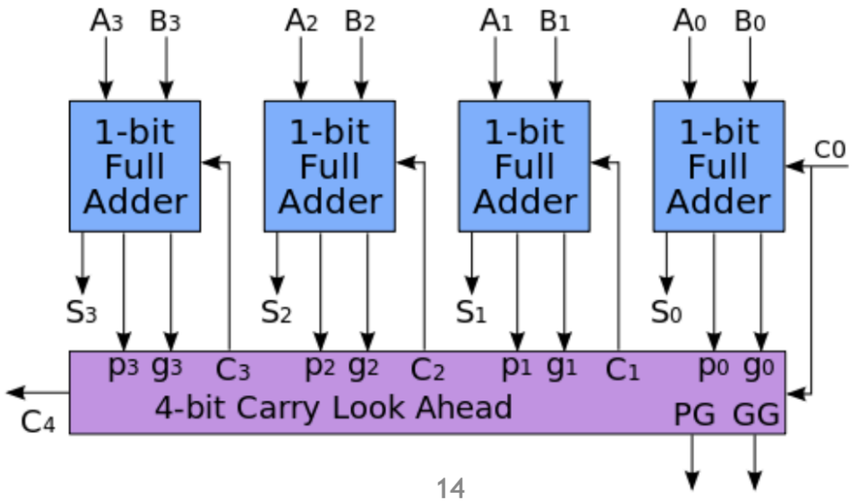
3 FPGA - 4-bit CLA

4 Lab Rule

Carry-Lookahead Adder (1/2)

Verilog Question 3

- (Gate Level) 4-bit Carry-Lookahead (CLA) Adder
  - Using your 1-bit Full Adder module in Lab 1 and logic gates
  - Please explain the benefits of a Carry-Lookahead Adder
  - Please explain **the circuit of your CLA and how it works**
- Go to Wikipedia to check out the details of it
  - https://en.wikipedia.org/wiki/Carry-lookahead_adder



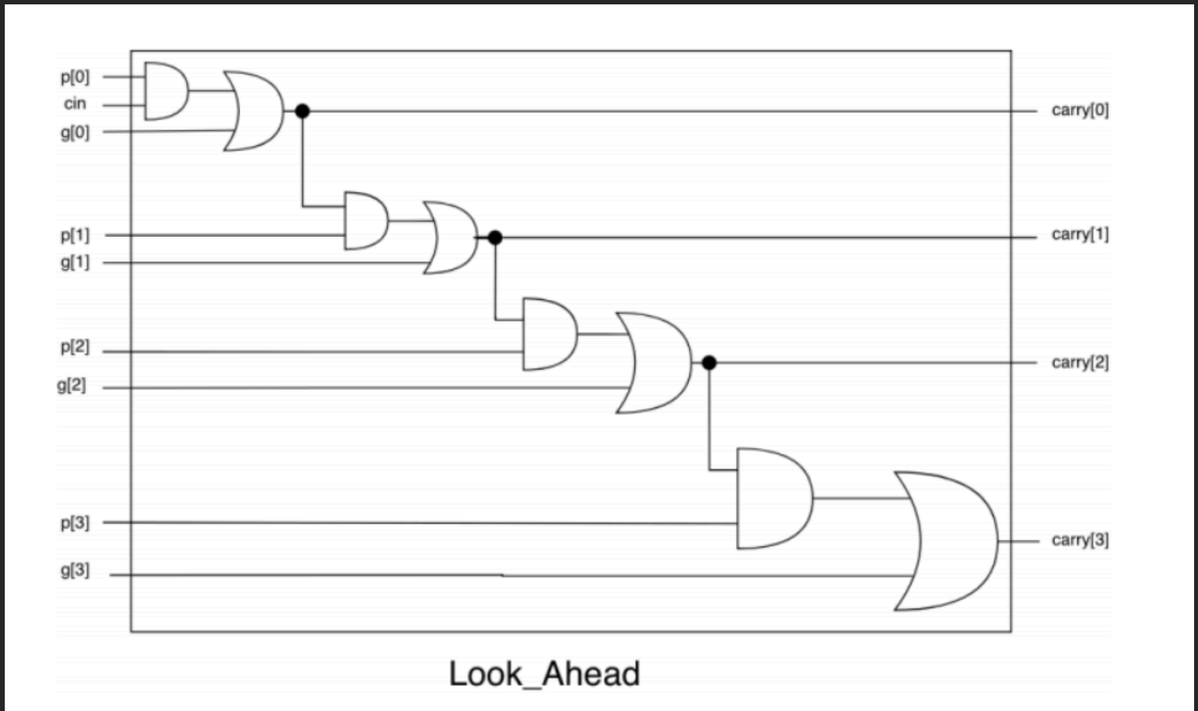
Basic - NOR gates only (1/2)

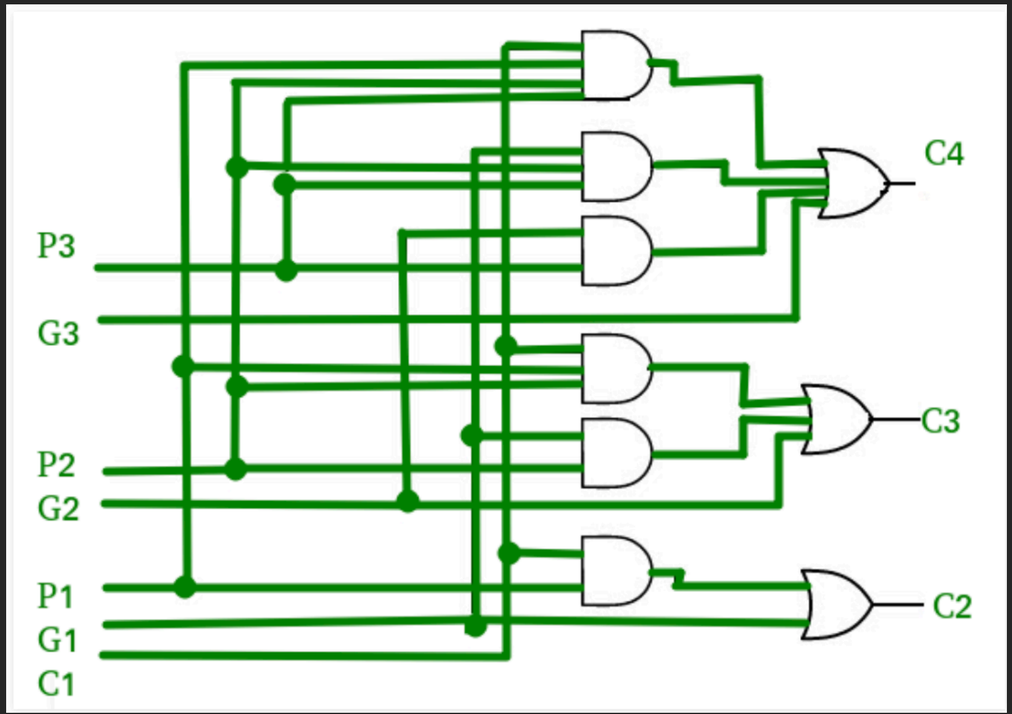
- 沒有解釋 Look Ahead module 的原理
- generate (g) and propagate (p)
- Circuit 設計錯誤 !!!!!!!!!!!!!!

kahoot.it

$$
\begin{aligned}
C_1 &= G_0 + P_0 \cdot C_0, \\
C_2 &= G_1 + G_0 \cdot P_1 + C_0 \cdot P_0 \cdot P_1, \\
C_3 &= G_2 + G_1 \cdot P_2 + G_0 \cdot P_1 \cdot P_2 + C_0 \cdot P_0 \cdot P_1 \cdot P_2, \\
C_4 &= G_3 + G_2 \cdot P_3 + G_1 \cdot P_2 \cdot P_3 + G_0 \cdot P_1 \cdot P_2 \cdot P_3 + C_0 \cdot P_0 \cdot P_1 \cdot P_2 \cdot P_3.
\end{aligned}
$$

Basic - NOR gates only (2/2)





$$
\begin{aligned}
C_1 &= G_0 + P_0 \cdot C_0, \\
C_2 &= G_1 + G_0 \cdot P_1 + C_0 \cdot P_0 \cdot P_1, \\
C_3 &= G_2 + G_1 \cdot P_2 + G_0 \cdot P_1 \cdot P_2 + C_0 \cdot P_0 \cdot P_1 \cdot P_2, \\
C_4 &= G_3 + G_2 \cdot P_3 + G_1 \cdot P_2 \cdot P_3 + G_0 \cdot P_1 \cdot P_2 \cdot P_3 + C_0 \cdot P_0 \cdot P_1 \cdot P_2 \cdot P_3.
\end{aligned}
$$

Outline

1 Basic - NOR gates only

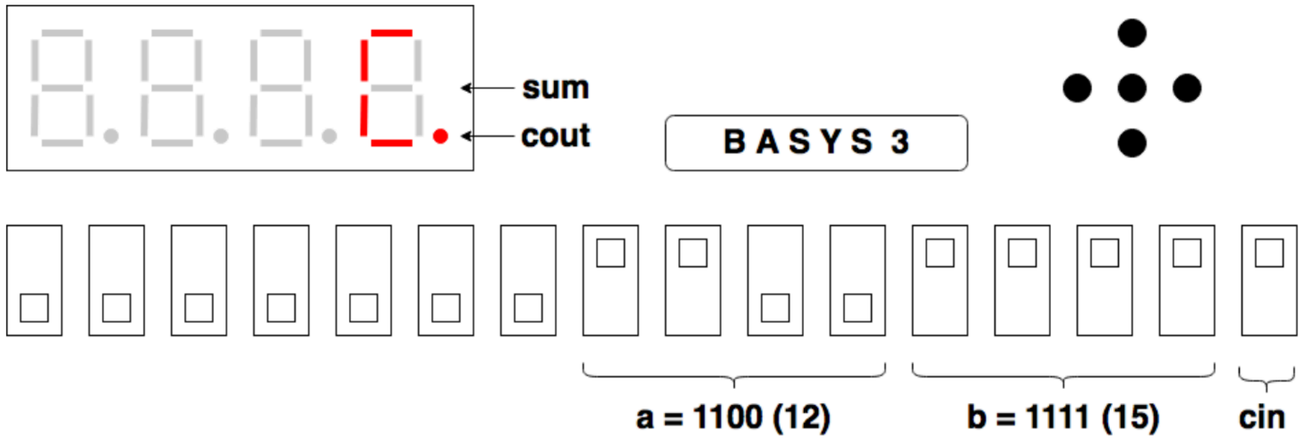
2 Advanced - Carry-Lookahead Adder

3 FPGA - 4-bit CLA

4 Lab Rule

FPGA - 4-bit CLA (1/2)

FPGA Demonstration 1



- (Gate Level) 4-bit Carry-Lookahead (CLA) Adder
- Implement a carry-lookahead adder to compute (**a + b**) and represent the sum in a **single hexadecimal number**
  - Please assign your inputs/outputs as:
    - **SW[0]** stands for '**cin**', **SW[8:5]** stands for '**a**', **SW[4:1]** stands for '**b**'
    - Use the **rightmost 7-segment display** to show your **sum**
    - Use the **rightmost dot** to show your **cout**

FPGA - 4-bit CLA (2/2)

- 沒有說明如何實作 out signal
- 沒有解釋如何設計 FPGA 的 top module，或是說明不夠詳細
- 請以文字解釋，不要直接貼code就算了
- 請畫出 top module 的 block diagram

Outline

1 Basic - NOR gates only

2 Advanced - Carry-Lookahead Adder

3 FPGA - 4-bit CLA

4 Lab Rule

Lab Rules (1/1)

- **Only gate-level description is permitted**
  - Only basic logic gates are ALLOWED (**AND, OR, NAND, NOR, NOT**)
  - **Sorry, no xor & xnor**

- **Please AVOID using**
  - Continuous assignment (e.g., **assign =**, **wire =**) and conditional operators (e.g., **: ?**)
  - Behavioral operators (e.g., **!**,**%**, **&**, ***, **+**, **/**, **<**, **>**, **^**, **|**, **~**)